

Field-based Coordination for Pervasive Computing Applications

Marco Mamei¹ and Franco Zambonelli¹

Dipartimento di Scienze e Metodi dell'Ingegneria,
University of Modena and Reggio Emilia
Via Allegri 13, 42100 Reggio Emilia, Italy
{mamei.marco, franco.zambonelli}@unimo.it

Abstract. Emerging pervasive computing technologies such as sensor networks and RFID tags can be embedded in our everyday environment to digitally store and elaborate a variety of information. By having application agents access in a dynamic and wireless way such distributed information, it is possible to enforce a notable degree of context-awareness in applications, and increase the capabilities of interacting with the physical world. In particular, biologically inspired field-based data structures such as gradients and pheromones are suitable to represent information in a variety of pervasive computing applications. This paper discusses how both sensor networks and RFID tags can be used to that purpose, outlining the respective advantages and drawbacks of these technologies.

keywords: Field-based coordination, Ad-hoc networks, RFID infrastructures, Bio-inspired computing

1 Introduction

Environment-mediated interaction (aka stigmergic interaction [1]) plays an important role in nature. Indeed, the spreading and sensing of pheromones in an environment to organize the activities of ant colonies, the process of morphogenesis as enforced by diffusion of chemicals in the embryo, the movement of masses induced by gravitational fields, are all examples of stigmergic interactions [2]. In the last few years, however, stigmergic models of interactions have been recognized as very powerful to facilitate interactions in dynamic distributed systems. Indeed, stigmergic models of interactions, whether relying on synthetic pheromones, on diffusion of digital chemicals, or on spreading of virtual computational fields, are being proposed to facilitate the enforcement of adaptive interaction patterns in dynamic distributed systems and to promote self-organization and self-adaptation of activities [1, 3, 4].

In the case of agents situated in a computational environment (e.g., the Web, a P2P network, or the Grid), supporting the interaction of agents with such an environment is a rather natural process. Simply, multiagent systems are computational entities the same as the environment, and once proper data formats

and interaction protocols are established, the access to the computational environment (and possibly the exploitation of such environment as an infrastructure in which to store the units of stigmergic interactions) becomes rather easy: the “sensors” and the “effectors” that the agents may use to interact reduce to a set of APIs or programming constructs.

The problem is totally different in the case of a physical environment. In this case, to access the physical environment, agents must be somehow be capable of perceiving and affecting physical properties. To this extent, an agent (whether in the form of an autonomous robot, or of an embedded controller, or of some software running on a mobile devices) must be necessarily supported by some hardware sensors and effectors to properly interact with the world.

Traditionally, most approaches for physically situated agents, assume that agents are augmented with the necessary capabilities for sensing and effecting the physical world. For instance, in the case of autonomous robots, traditional approaches assume that the robot itself is equipped with video-cameras, temperature sensors, location sensors (e.g., GPS), and robotic hands. Such approach tends to notably increase the internal complexity of agents. In fact, agents not only have to perform the computational activities associated to deciding how to accomplish a goal, but have also to take care of properly internalizing and interpreting the data coming from the associated sensors, and of properly controlling their effectors to actualize their actions.

Another drawback of the above approach is that the physical environment can hardly be used to support stigmergic models of interactions, unless one adopt rather tricky solutions. If the environment is purely physical, in fact, stigmergic interactions should occur by physically affecting the properties of the environment. For example, to mimic the behavior of ants, robots would be forced to actually pollute the environment with some kind of marker, and would have to be equipped with sensor to perceive such marks [5].

The advent of pervasive computing technologies dramatically changes this scenarios. The availability of small-scale and low-cost devices that can be distributed in physical environment in a non intrusive way, that can be devoted to sense (or affect) specific properties in the environment, and that enable to interact with them in a wireless way (a capability to be easily provided to agents), enables agents to externalize all the activities devoted to interpret and control their physical activities. Simply, sensing and effecting the environment reduces in properly accessing some digital services. The result is in a notable reduction of complexity in agents, both at the hardware and at the software level.

In addition, the presence in an environment of embedded computational resources, as those that can be provided by the embedded computing devices, can be fruitfully exploited as an infrastructure to support stigmergic models of interactions. In fact, stigmergy can take place without actually affecting the physical environment, but simply by exploiting the distributed embedded resources as stores for those data structures that are at the basis of stigmergy, e.g., pheromones, fields, etc.

Clearly, depending on the specific technologies and devices adopted, the interactions with the environment and the support of stigmergic coordination models can be more or less facilitated. In the following of this paper, we analyze in detail two different classes of devices (sensor networks (in Sect. 2) and RFID tags (in Sect. 3), discuss how they can be exploited, and outline their respective advantages and drawbacks.

2 Ad-Hoc and Sensor Network

Future pervasive computing scenarios comprise a huge number of heterogeneous devices interacting with each other to achieve complex distributed applications. Sensor networks and networks of handheld computers could be employed in a variety of applications including environmental monitoring [6, 7], navigation [8], and human interaction support [9].

In general terms, sensor networks are an ideal platform to augment the physical environment with digital information.

- Sensors can store data to represent some kind of contextual information. Moreover, they can deliver such data to agents (e.g., users with PDA) passing nearby.
- Sensors can perform computations to support and facilitate the agents' fruition to that data. For example, sensors can propagate and diffuse data across the network. They can automatically delete old and possibly corrupted information. They can combine and transform data to let it become more expressive and easy to use.

Other than providing contextual information coming from the “outside” world, sensor network can also be used to store and convey information produced by the agent themselves. Moreover, relying on the sensor networking capabilities it is possible to spread distributed data structures across the environment.

In particular, we can imagine that each component of the system (software agent, wireless device, embedded sensor, etc.) is capable of generating and propagating field-like data structures that convey some information about their context. Agents can perceive these fields and react accordingly. The idea is that components are simply driven by these force fields as if they were particles under the action of a gravitational field.

Field-based data structures are distributed data structures encoding specific aspects of the application components' operational environment. These fields are propagated across a network by a component in order to represent and “communicate” its own activities. Field data structures are easily accessible by the components and provide easy-to-use context information (i.e., the overlays are specifically conceived to support their access and fruition).

The strength of these overlay data structures is that they can be accessed piecewise as the application components visit different places of the distributed environment. This lets the components to access the right information at the

right location. In addition, overlay data structures decouple components' activities from the underlying network dynamism. Components interacting and perceiving their operational environment by means of these overlay data structure can disregard the underlying physical network and its dynamics.

To clarify these concepts let us focus on the problem of coordinating the movements of some autonomous components (i.e., agents) in a distributed environment [10]. Hereafter we will use the term agent to refer to any autonomous real-world or software entity with computing and networking capability (e.g., a user carrying on a Wi-Fi PDA, a robot, or a modern car). In particular, we focus on the simple application of having two persons, provided with a PDA, moving across an environment instrumented with an ad-hoc network infrastructure. The goal of the application is to allow one person to be guided by the PDA, to follow the other person. A simple solution based on overlay data structures is the let the person to-be-followed to spread in the environment (i.e., ad-hoc network) a data structure that increases an integer value by one at every hop as it gets farther from the source. This creates a sort of gradient that can be followed downhill by the other person to complete the application [10] (see Fig. 1(a)). If the person to-be-followed moves, it is important that the overlay data structure adjust its shape accordingly, so that the gradient leads to that person anyway (see Fig. 1(b)). The power of this approach is that the overlay data structure provides expressive contextual information tailored for that specific task. The agent running on the PDA does not need to know any map of the environment, nor it has to execute complex algorithms to decide where to go. It just blindly follows the overlay data structure.

Beside this exemplary application, overlay data structures are general purpose and can be applied in a wide range of application scenarios, ranging from robotics to network routing [10, 11].

2.1 Pros and Cons

The power of this approach is that the distributed data structure provides expressive contextual information tailored for that specific task. The agent running on the PDA does not need to know any map of the environment, nor it has to execute complex algorithms to decide where to go. It just blindly follows the field data structure. All the complexity of the application is moved away from the agents and diverted into the environment-infrastructure.

Sensor networks are a powerful technology to support environment abstractions in multi-agent systems. In the long run, once current technological problems will be properly addressed, it will be the leading infrastructure of environment applications. Its main strength is that it is an *active* infrastructure: sensor nodes can run (distributed) algorithms to process data as required. For example, sensor nodes can proactively delete old information or run algorithm to aggregate data on needs. At present, however, this is also sensor network main weakness. Nodes suffer, in fact, from battery-exhaustion problems, they are costly and failure prone.

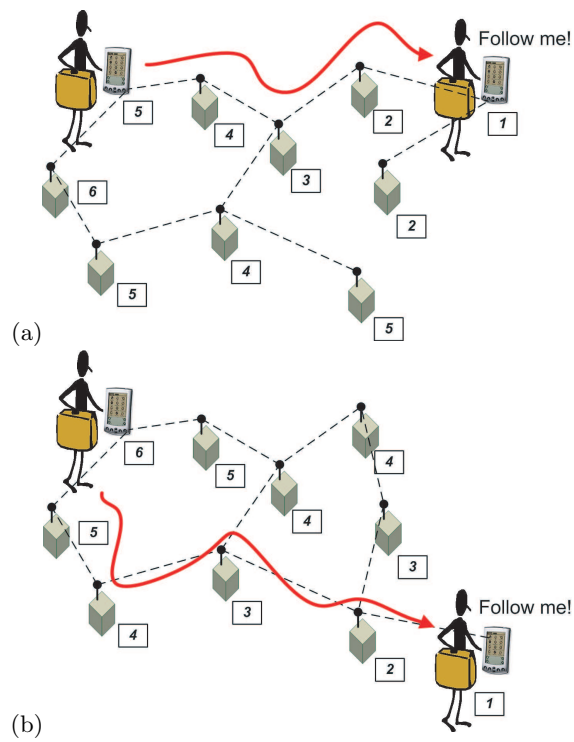


Fig. 1. (a) A gradient data structure enables an agent to follow another one. (b) The data structure is updated to reflect the new agent position.

2.2 Related Work

A number of recent proposals address the problem of defining supporting environments for the development of adaptive, dynamic, context-aware distributed applications, suitable for pervasive computing.

The TinyLime middleware [12] proposes accessing the environmental data collected by a sensor network via an associative tuple-based mechanisms. When a user with a mobile device “walks-through” a network of distributed sensors, all the data collected by the in-range sensors automatically feeds a local tuple space of the mobile device, which thus can perceive sensorial data collected by sensors simply by reading in the local tuple space.

ObjectPlaces [13] is an interesting middleware infrastructure that offers support to exchange and share information among nodes in mobile and ad-hoc networks. Specifically, in ObjectPlaces, agents communicate indirectly through the exchange of objects that can be temporarily stored across suitable object-places (that are virtual containers stored in the ad-hoc network itself). Agents invoke operations to add and remove objects, or to observe the content of a specific object-place (via a pattern-matching process). Agents can also create object-places dynamically, and link them together to form a graph-like environment connecting related object-places.

TOTA [10] and Smart Messages [14] are two architectures for computation and communication in large networks of embedded systems. Communication is realized by sending “smart tuples” in the network, i.e., tuples which include code to be executed at each hop in the network path. These models comply with the general idea of putting intelligence in the network by letting tuples and messages execute hop-by-hop small chunk of code to determine their propagation.

Lime [15] and XMIDDLE [16] exploits transiently tuple spaces as the basis for interaction in dynamic network scenario. Each mobile device, as well as each network nodes, owns a private tuple space. Upon connection with other devices or with network nodes, the privately owned tuple spaces can merge in a federated tuple space, to be used as a common data space to exchange information.

3 RFID Technology

Advances in miniaturization and manufacturing have yielded postage-stamp sized radio transceivers called Radio Frequency Identification (RFID) tags that can be attached unobtrusively to objects as small as a toothbrush. The tags are wireless and battery free. Each tag is marked with a unique identifier and provided with a tiny memory, up to some KB for advanced models, allowing to store data. Tags can be purchased off the shelf, cost roughly 0.20 Euro each and can withstand day-to-day use for years (being battery-free, they do not have power-exhaustion problems). Suitable devices, called RFID readers, can access RFID tags by radio, either for read and write operations. The tags respond or store data accordingly using power scavenged from the signal coming from the RFID reader. RFID readers divide into short- and long-range depending on the

distance within which they can access RFID tags. Such distance may vary from few centimeters up to some meters. Deploying RFID technology requires that a number of places in the environment (e.g. doors, corridors, etc.) or objects (e.g. beds, washing machines, etc.) are tagged with RFID tags. Tagging a place or an object involves sticking an RFID tag on it, and making a database entry mapping the tag ID to a name. It is worth emphasizing that current trends indicate that within a few years, many household objects and furniture may be RFID-tagged before purchase, thus eliminating the overhead of tagging [17]. Moreover, some handheld devices start to be provided with RFID read and write capabilities (the Nokia 5140 phone can be already equipped with a RFID reader [18]).

The set of RFID tags deployed across the environment can be regarded as an infrastructure to store and deliver digital information.

From a general perspective, accessing the RFID tags nearby is a powerful source of context information. For example, RFID tags can reveal the location of agents in that tags can be associated to uniquely identified places. So reading the tag associated with “Prof. Smith desk” can let an agent infer its location as “Prof. Smith office”. More in general, the knowledge of RFID tags (and thus objects) nearby can possibly identify a specific application context (e.g. reading a LCD-projector tag, and a microphone tag can let the agent infer of being in a meeting room).

In addition, given the fact that RFID tags can be written on-the-fly, agents can use the tags as a distributed shared memory with which to exchange information. For example, RFID tags can be accessed as if they were distributed tuple spaces [19, 20]. A particularly significant development of this idea is related to spreading pheromone-inspired distributed data structures across the tags in the environment. The basic scenario consists of human users and robots carrying handheld computing devices, provided with a RFID reader, and running an agent-based application. The agent, unobtrusively from the user, continuously detects in range tags as the user roams across the environment. Moreover, the agent controls the RFID reader to write pheromone data structures (consisting at least in a pheromone ID) in all the tags encountered. This process creates a digital pheromone trail distributed across the tags. More formally, let us call $L(t)$ the set of tags being sensed at time t . It is easy to see that the agent can infer that the user is moving if $L(t) \neq L(t-1)$ (see Fig. 2).

If instructed to spread pheromone O , the agent will write O in all the $L(t)$ - $L(t-1)$ tags as it moves across the environment. For the majority of applications a pheromone trail, consisting of only an ID, is not very useful. Like in ant foraging, most applications involve agents to follow each other pheromone trails to reach the location where the agents that originally laid down the trail were directed (or, on the contrary, to reach the location where they came from). Unfortunately, an agent crossing an-only-ID-trail would not be able to choose in which direction the agent that laid down that trail was directed. From the agent point of view, this situation is like crossing a road without knowing whether to turn left or right.

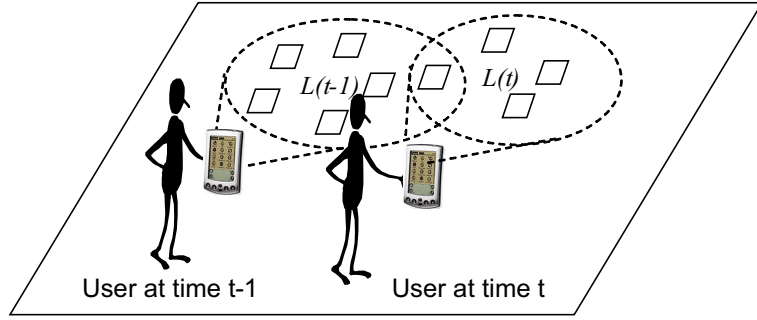


Fig. 2. When the user moves, its agent gets in range with a different set of location-tags (here represented as white rectangles), and recognizes the motion.

To overcome this problem, the agent stores in the tags also an ever increasing hop-counter associated with O - we will call this counter $C(O)$. In particular, if an agent decides to spread pheromone O at time t , the agent reads also the counter $C(O)$ in the $L(t)$ set (if $C(O)$ is not present, the agent sets $C(O)$ to a fixed value zero). Upon a movement, the agent will store O and $C(O)+1$ in the tags belonging to $L(t+1)$ that do not have O or have a lower $C(O)$. In addition, the basic pheromone idea requires a pheromone evaporation mechanism to discard old - possibly corrupted - trails. To this end we store in the tag also a value $T(O)$ representing the time where the pheromone O has been stored. (see Fig. 3)

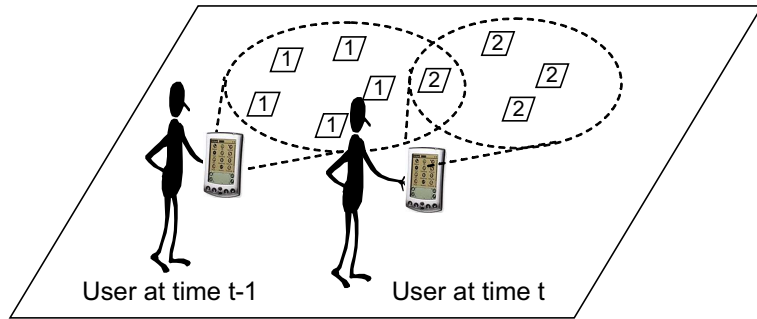


Fig. 3. Writing of pheromone information in RFID tags.

To read pheromones, an agent trivially accesses neighbor RFID tags reading their memories. Since RFID read operations are quite unreliable, the agent actually performs a reading cycle merging the results obtained at each iteration. Given the result, the agent will decide how to act on the basis of the perceived pheromone configuration. To realize pheromone evaporation, after reading a tag,

an agent checks, for each pheromone, whether the associated timestamp is, accordingly to the agent local time, older than a certain threshold T . If it is so, the agent deletes that pheromone from the tag. This kind of pheromone evaporation leads to two key advantages:

1. Since the data space in RFID tags is severely limited, it would be most useful to store only those pheromone trails that are important for the application at a given time; old, unused pheromones can be removed.
2. If an agent does not carry its personal digital assistant or if it has been switched off, it is possible that some actions will be undertaken without leaving the corresponding pheromone trails. This cause old-pheromone trails to be possibly out-of-date, and eventually corrupted. In this context, it is of course fundamental to design a mechanism to reinforce relevant pheromones not to let them evaporate.

With this regard, an agent spreading pheromone O , will overwrite O -pheromones having an older $T(O)$. From these considerations, it should be clear that the threshold T has to be tuned for each application, to represent the time-frame after which the pheromone is considered useless or possibly corrupted.

3.1 Pros and Cons

The main point in favor of this approach is its extremely low cost since it uses technologies (RFID) that are likely to be soon embedded in the scenario independently of this application. Relying on such an implementation, a wide range of application scenarios based on pheromone interaction can be realized ranging from multi-robot coordination [21], to monitoring of human activities [22].

The main problem with this approach is related to current limitations of RFID technology. Accessing tags for reading and writing operations can fail for a number of hardly controllable issues (electromagnetic disturbances, metallic objects nearby, interferences and collisions, etc.). Moreover, in the next section, we will present and discuss some limitations in our RFID implementation of the pheromone evaporation mechanism.

3.2 Related Work

Several proposals, as well as ours, consider the idea of having mobile devices integrated with a RFID reader, thus having the capability of accessing RFID tags around, as sorts of digital contextual information stores. However, rather than considering the possibility of storing new information in RFID tags and enforcing coordination through them, most approaches exploit RFID tags only for reading pre-existent environmental/contextual information. For instance, the system described in [23] proposes associating location information with tags (e.g., "I am the tag of the living room") that can be read by mobile robots carrying on a RFID reader to roughly localize themselves.

The system described in [22] exploits RFID tags for inferring information about contextual activity in an environment. Users are assumed to wear an RFID reader connected with a Wi-Fi portable device so that, when the user moves and acts in the environment, the type and the sequence of tags read by the reader can suggest what the user is doing. For example, reading the tag associated to the user boss and of a video projector can let infer that the user is in a sort of important meeting with his/her boss

Pheromones spread in the environment can enable a group of users (both humans and robotics) to coordinate their respective movements. An exemplary application would be distributed environment exploration. Users could decide to explore a specific area if there are not pheromones pointing in that direction (the area is truly unexplored). In this context, it is important to remark that this approach clearly requires the presence of RFID tags before pheromones can be spread. If the environment does not contain tags at all, this approach could not be used. However, on the one hand, RFID tags are likely to be soon densely present in everywhere (embedded in tiles, bricks, furniture, etc.). On the other hand, it is possible to conceive solutions where agents physically deploy RFID tags while exploring the environment to be used for subsequent coordination. For instance, future development in plastic (and printable) RFID technology [24] let us envision the possibility of enriching an agent with a simple RFID printer to dynamically print in pavements, walls, or any type of surface, RFID tags.

4 Conclusion and Future Work

This paper presented the role of sensor network and RFID-based infrastructures to support environment abstraction and field-based coordination in pervasive computing scenarios. These infrastructures not only allow agents to acquire context information, but also can serve as suitable media to support their coordination activities.

Our future work in this direction is twofold. On the one hand, we will try to solve technological problems related to current hardware limitations. On the other hand, we will try to apply such mechanisms and abstractions to several pervasive computing scenarios.

Acknowledgments. Work supported by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission.

References

1. Parunak, V.: Go to the ant: Engineering principles from natural agent systems. *Annals of Operations Research* **75** (1997) 69–101
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press, Oxford (UK) (1999)
3. Babaoglu, O., Meling, H., Montresor, A.: A framework for the development of agent-based peer-to-peer systems. In: *22nd International Conference on Distributed Computing Systems*. IEEE CS Press, Vienna, Austria (2002) 15 – 22

4. Mamei, M., Zambonelli, F.: Physical deployment of digital pheromones through rfid technology. In: IEEE Swarm Symposium. IEEE CS Press, Pasadena (CA), USA (2005)
5. Svennebring, J., Koenig, S.: Building terrain covering ant robots: a feasibility study. *Autonomous Robots* **16** (2004) 313 – 332
6. Paskin, M., Guestrin, C., McFadden, J.: A robust architecture for inference in sensor networks. In: International Symposium on Information Processing in Sensor Networks. ACM Press, Los Angeles (CA), USA (2005)
7. Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., Welsh, M.: Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing* **10** (2004) 18 – 25
8. Patterson, D., Liao, L., Fox, D., Kautz, H.: Inferring high-level behavior from low-level sensors. In: International Conference on Ubiquitous Computing. ACM Press, Seattle (WA), USA (2003)
9. Choudhury, T., Pentland, A.: Sensing and modeling human networks using the sociometer. In: International Symposium on Wearable Computers. IEEE CS Press, White Plains (NY), USA (2003)
10. Mamei, M., Zambonelli, F.: Programming pervasive and mobile computing applications with the tota middleware. In: Proceedings of the International Conference On Pervasive Computing (Percom). IEEE CS Press, Orlando, Florida, USA (2004)
11. Stoy, K., Nagpal, R.: Self-reconfiguration using directed growth. In: 7th International Symposium on Distributed Autonomous Robotic Systems. Springer-Verlag, Toulouse, France (2004)
12. Curino, C., Giani, M., Giorgetta, M., Giusti, A., Murphy, A., Picco, G.: Tinylime: Bridging mobile and sensor networks through middleware. IEEE CS Press (2005)
13. Weyns, D., Schelfhout, K., Holvoet, T.: Exploiting a virtual environment in a real-world application. In: Proceedings of the International Workshop on Environments for Multiagent Systems, Utrecht, NL (2005)
14. Riva, O., Nadeem, T., Borcea, C., Iftode, L.: Context-aware migratory services in ad hoc networks. *IEEE Transaction on Mobile Computing* (to appear) (2007)
15. Picco, G., Murphy, A., Roman, G.: Lime: a coordination model and middleware supporting mobility of hosts and agents. *ACM Transactions on Software Engineering and Methodology* **15** (2006) 279–328
16. Mascolo, C., Capra, L., Zachariadis, Z., Emmerich, W.: Xmiddle: A data-sharing middleware for mobile computing. *Wireless Personal Communications* **21** (2002) 77 – 103
17. (Smart-Mobs) <http://www.smartmobs.com>.
18. (Nokia-Mobile-RFID-Kit) <http://www.nokia.com/nokia/0,,55738,00.html>.
19. Mamei, M., Zambonelli, F.: Pervasive pheromone-based interaction with rfid tags. *ACM Transactions on Autonomous and Adaptive Systems* **2** (2007) 1–28
20. Mamei, M., Quaglieri, R., Zambonelli, F.: Making tuple spaces physical with rfid tags. In: Proceedings of the Symposium on Applied Computing (SAC). ACM Press, Dijon, France (2006)
21. Payton, D., Daily, M., Estowski, R., Howard, M., Lee, C.: Pheromone robotics. *Autonomous Robots* **11** (2001) 319 – 324
22. Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., Hahnel, D.: (Inferring activities from interactions with objects)
23. Kulyukin, V., Gharpure, C., Nicholson, J., Pavithran, S.: Rfid in robot-assisted indoor navigation for visually impaired. In: Proceedings of the International Conference on Intelligent Robots and Systems. IEEE CS Press (2004)
24. Collins, G.: Next stretch for plastic electronics. *Scientific American* (2004)